Parallel computation of interfacial dynamics(Domain Decomposition Methods and Related Topics)

AUTHOR(S):

IKEDA, TSUTOMU; UEDA, AKIHO

CITATION:

IKEDA, TSUTOMU ...[et al]. Parallel computation of interfacial dynamics(Domain Decomposition Methods and Related Topics). 数理解析研究所講究録 1997, 989: 68-77

ISSUE DATE:

1997-04

URL:

http://hdl.handle.net/2433/61071

数理解析研究所講究録
989巻 1997年 68-77

Parallel computation of interfacial dynamics

TSUTOMU IKEDA AND AKIHO UEDA

Department of Applied Mathematics and Informatics,
Ryukoku University,
Seta Ohtsu 520-21, Japan
e-mail: tsutomu@rins.ryukoku.ac.jp

Abstract: Parallel computers become available now in a lot of research institutes and they will take the most important part in scientific computation instead of vector computers in near future. ¿From theoretical point of view, on the other hand, the domain decomposition method has been carefully studied. Under this situation, we planed to develop a parallel computation scheme for reaction-diffusion phenomena based on the domain decomposition method and a preconditioned conjugate gradient method.

1 Introduction

We are concerned with the pattern evolution far from equilibrium, and treat of the dynamics of sharp transition internal layers arising from reaction-diffusion systems. Reaction-diffusion systems have been utilized for describing various nonlinear phenomena and have contributed to understanding the mechanism of pattern formation in wide fields of natural sciences. Many mathematicians have developed advanced analytical tools for studying the pattern evolution and for tracking its asymptotic behavior. Rigorous treatments, however, are not sufficient for this purpose except a few cases. It is hard to extract essential features of layer dynamics by investigating the full system of nonlinear equations in mathematically rigorous way.

One of reasonable approaches is to track the pattern formation and evolution by numerical simulation using high speed computers. You may find completely new behavior of solutions through numerical computation. You may also find a new rigorous approach and get a method of proof by checking numerical results obtained by computer simulation.

Parallel computers become available now in a lot of research institutes and universities. Because of their potential for both high-performance and cost-effectiveness parallel computers will attract much more attention of researches, and they will take the most important part in scientific computation instead of vector computers in near future. ¿From theoretical point of view, on the other hand, the domain decomposition method has been carefully studied. Under this situation, we planed to develop a parallel computation scheme for a system of reaction-diffusion equations based on the domain decomposition method and a preconditioned conjugate gradient method.

[1]This work is supported in part by Joint Research Center for Science and Technology, Ryukoku University.

2 Numerical computation of reaction-diffusion systems

In the present paper, we deal with the following general type of reaction-diffusion systems in two-dimensional space:

$$\begin{aligned} \frac{\partial u}{\partial t} &= d_u \Delta u + f(u,v) \\ \frac{\partial v}{\partial t} &= d_v \Delta v + g(u,v) \end{aligned} \qquad \text{in } \Omega \subset \mathbf{R}^2 \text{ and } t > 0, \tag{2.1}$$

where Ω is a bounded domain, Δ denotes the Laplace operator, d_u and d_v are diffusion coefficient of u and v, respectively. The nonlinear terms $f(u,v)$ and $g(u,v)$ express the reaction effects between u and v. The unknown variables $u(t,x,y)$ and $v(t,x,y)$ stand for the density of biological species in a case and the density of chemical substances in another case. So the homogeneous Neumann boundary condition

$$\frac{\partial u}{\partial \mathbf{n}} = 0 \quad \text{and} \quad \frac{\partial v}{\partial \mathbf{n}} = 0 \qquad \text{on } \partial\Omega \tag{2.2}$$

is suitable for the unknowns in many models, where $\mathbf{n}$ denotes the unit outer normal vector on the boundary $\partial\Omega$ of Ω.

Patterns are formed by a delicate balance between diffusion effects and reaction dynamics, and the time increment Δt may play a crucial role in the numerical computation for some reaction-diffusion systems. In fact, subtle patterns like traveling breathers ([3]) are reproduced only for a very narrow range of Δt. By this reason many researchers prefer the implicit approximation for the diffusion terms, and choose the explicit one for the nonlinear reaction terms to avoid unpredictable iterations:

$$\begin{aligned} \frac{1}{\Delta t}(u^{n+1} - u^n) &= d_u \Delta u^{n+1} + f(u_n, v_n) \\ \frac{1}{\Delta t}(v^{n+1} - v^n) &= d_v \Delta v^{n+1} + g(u_n, v_n) \end{aligned} \qquad \text{in } \Omega \subset \mathbf{R}^2 \text{ and } n = 0, 1, \cdots. \tag{2.3}$$

Thus the numerical computation of (2.1) with (2.2) is to solve a sequence of the following Helmholtz type equations subject to the zero-flux boundary condition:

$$u - d\Delta u = f(x,y) \text{ in } \Omega, \quad \frac{\partial u}{\partial \mathbf{n}} = 0 \text{ on } \partial\Omega. \tag{2.4}$$

3 Domain decomposition method

There are two types of parallel computations for (2.4):

(1) one discretizes (2.4) by the conventional finite element method, finite difference method or boundary element method, and devises a parallel code solving the resulting system of linear equations,

(2) one derives a new formulation of (2.4) that permits to devise an effective parallel algorithm easily.

A parallel code of type (1) and its corresponding original serial code give the same results of course, however, numerical solutions of parallel algorithm of type (2) are not always the same as those obtained by a conventional serial scheme. In this article, employing the non-overlapping domain decomposition method, which is a typical approach belonging to the type (2), we propose a parallel computation scheme for (2.4).

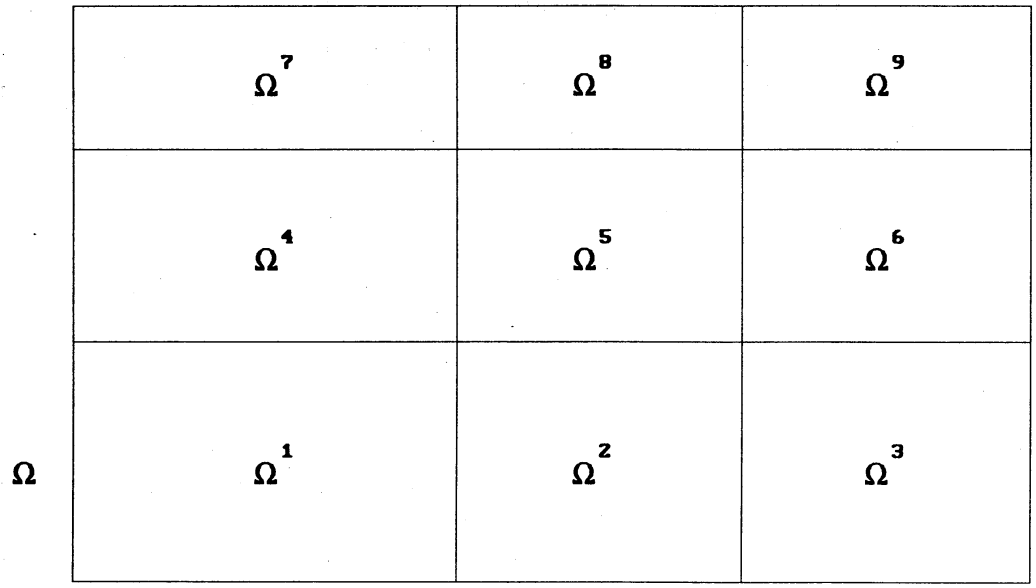

Figure 3.1: Domain decomposition into 9 non-overlapping subdomains

We explain our parallel computation scheme for (2.4) by assuming Ω to be a rectangle for simplicity. We decompose Ω into N non-overlapping (rectangular) subdomains as shown in Figure 3.1. The closure of additional boundary of sub-rectangle Ω^m contained in the interior of Ω is denoted by Γ^m = closure of $(\partial\Omega^m \setminus \partial\Omega)$, $m = 1, \cdots, N$. We put

$$\Xi = \{(m,n) : m \neq n,\ \Omega^m \text{ and } \Omega^n \text{ have the same side}\},$$

and denote internal interfaces between sub-rectangles by $\Gamma^{mn} = \Gamma^m \cap \Gamma^n$, $(m,n) \in \Xi$. The decomposition of domain does not enable us to consider (2.4) independently in each subdomain. The decomposition imposes artificial boundary conditions on internal interfaces, but it increases very much the independence level of computation in subdomains nevertheless. Adopting fluxes of $u(x,y)$ as unknown boundary data ([4]), we define the following weak form of (2.4) in our domain decomposition method:

Find $u^m \in H^1(\Omega^m)$ and $\lambda^{mn} \in H^{-1/2}(\Gamma^{mn})$, $m = 1, \cdots, N$ and $(m,n) \in \Xi$, such that

$$\int_{\Omega^m} (u^m v^m + d\nabla u^m \nabla v^m) - \sum_{(m,n)\in\Xi} \int_{\Gamma^{mn}} \lambda^{mn} v^m = \int_{\Omega^m} f v^m \quad \forall v^m \in H^1(\Omega^m), \quad m = 1, \cdots, N, \tag{3.1}$$

$$\int_{\Gamma^{mn}} (u^m - u^n)\mu^{mn} = 0 \quad \forall \mu^{mn} \in H^{-1/2}(\Gamma^{mn}), \qquad (m,n) \in \Xi,$$

$$\lambda^{mn} + \lambda^{nm} = 0 \quad \text{on } \Gamma^{mn}, \qquad (m,n) \in \Xi.$$

Here, $\mathbf{n}^{mn}$ is the unit outer normal vector on the interface Γ^{mn} with respect to Ω^m and $\lambda^{mn} = d\dfrac{\partial u^m}{\partial \mathbf{n}^{mn}}$ is the flux of $u(x,y)$ on Γ^{mn} to be sought, $(m,n) \in \Xi$. The first equation in (3.1) is the weak form of (2.4) in each subdomain Ω^m. The second one requires the continuity of $u(x,y)$ on Γ^{mn} and the third one states only that $\mathbf{n}^{mn}$ and $\mathbf{n}^{nm}$ are in opposite direction.

Introducing an orthogonal lattice onto the rectangle Ω and adding diagonal lines, we decompose each subdomain Ω^m into finite elements (triangulation of Friedrichs-Keller type). Every finite element is a right triangle having sides parallel to the x- and y-axes. The unknown function u^m is approximated by a piecewise linear continuous function u_h^m on Ω^m, $m = 1, \cdots, N$, and the flux λ^{mn} is also approximated by a piecewise linear continuous function λ_h^{mn} on Γ^{mn}, $(m,n) \in \Xi$. In Figure 3.2 for instance, $\bullet$ expresses the lattice point where u_h^m or λ_h^{mn} has its value freely. We have to take a special care for lattice points corresponding to cross-points of sub-rectangles brought by the domain decomposition, that is,

(*) at every cross-point $\Gamma^{mnpq} = \Gamma^m \cap \Gamma^n \cap \Gamma^p \cap \Gamma^q$ ($\mathbf{n}^{mn} = \mathbf{n}^{pq}$ and $\mathbf{n}^{mp} = \mathbf{n}^{nq}$), three of λ_h^{mn}, λ_h^{mp}, λ_h^{nq} and λ_h^{pq} have their values freely but exactly one of them has to take the same value as that standing its opposite side.

At the cross-point $\Gamma^{1245} = \Gamma^1 \cap \Gamma^2 \cap \Gamma^4 \cap \Gamma^5$ in Figure 3.2 for instance, λ_h^{12}, λ_h^{25} and λ_h^{45} have their values freely, however, λ_h^{14} has to take the same value as λ_h^{25}. This restriction is derived by noting the fact that all of u_h^1, u_h^2, u_h^4 and u_h^5 have the same value at Γ^{1245} if and only if three of the following equalities hold:

$$u_h^1(\Gamma^{1245}) = u_h^2(\Gamma^{1245}), \quad u_h^2(\Gamma^{1245}) = u_h^5(\Gamma^{1245}),$$
$$u_h^1(\Gamma^{1245}) = u_h^4(\Gamma^{1245}) \text{ and } u_h^4(\Gamma^{1245}) = u_h^5(\Gamma^{1245}).$$

We also note that by this treatment the coefficient matrix $\mathbf{BRA}^{-1}\mathbf{R}^T\mathbf{B}^T$ in (5.1) becomes positive definite.

Let $X_h(D)$, $D = \Omega^m$ and $D = \Gamma^{mn}$, $m = 1, \cdots, N$ and $(m,n) \in \Xi$, be the space of piecewise linear continuous functions on D, and denote the mass lumping operator by $\bar{}$. Then, our finite element scheme for (3.1) is define by

Find $u_h^m \in X_h(\Omega^m)$ and $\lambda_h^{mn} \in X_h(\Gamma^{mn})$, $m = 1, \cdots, N$ and $(m,n) \in \Xi$, such that

$$\begin{aligned}
&\int_{\Omega^m} (\bar{u}_h^m \bar{v}_h^m + d\nabla u_h^m \nabla v_h^m) - \sum_{(m,n)\in\Xi} \int_{\Gamma^{mn}} \bar{\lambda}_h^{mn} \bar{v}_h^m = \int_{\Omega^m} \bar{f}_h \bar{v}_h^m \quad \forall v_h^m \in X_h(\Omega^m), \\
&\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad m = 1, \cdots, N, \\
&\int_{\Gamma^{mn}} (\bar{u}_h^m - \bar{u}_h^n)\bar{\mu}_h^{mn} = 0 \quad \forall \mu_h^{mn} \in X_h(\Gamma^{mn}), \qquad (m,n) \in \Xi, \\
&\lambda_h^{mn} + \lambda_h^{nm} = 0 \quad \text{on } \Gamma^{mn}, \qquad (m,n) \in \Xi,
\end{aligned} \tag{3.2}$$

restrictive condition on λ_h^{mn}'s corresponding to (*).

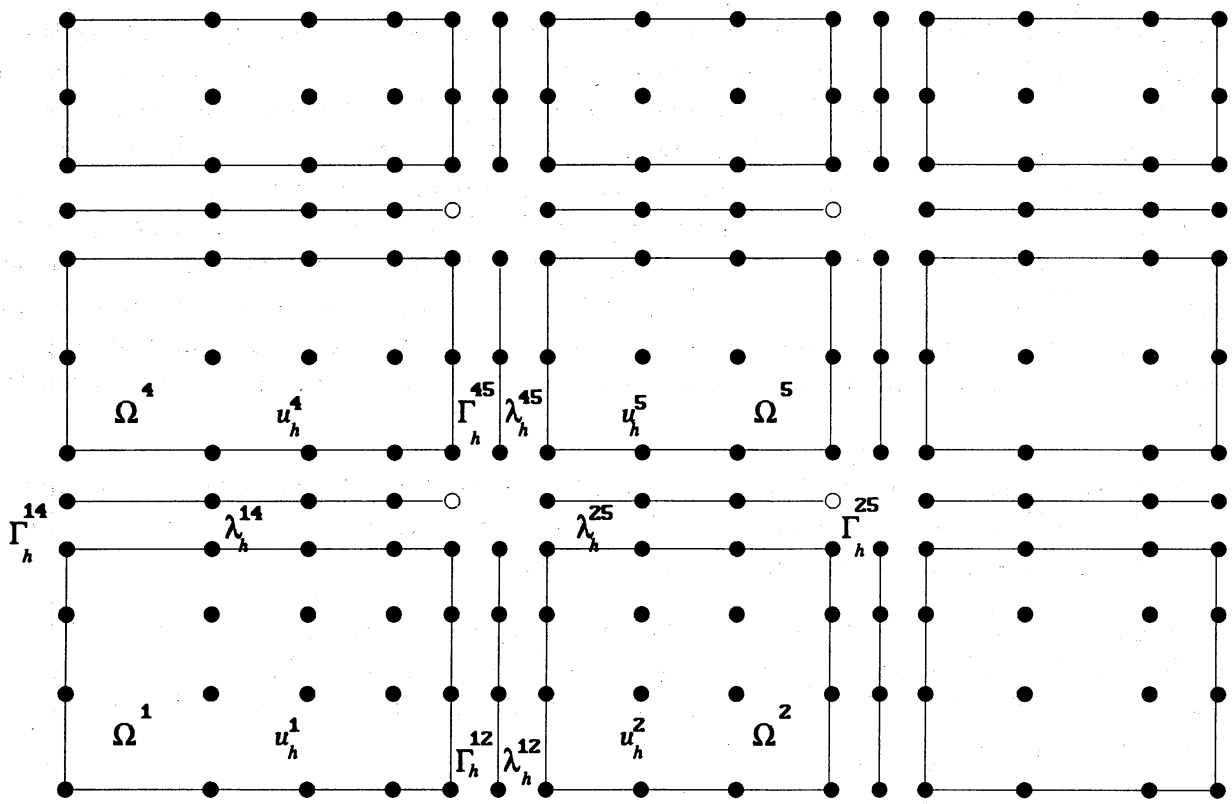

Figure 3.2: Nodal points for u_h^m and λ_h^{mn}

The last condition in (3.2) may look curious, however, we obtain:

Proposition 3.1 *(1) The problem (3.2) has a unique solution* $\{u_h^m, \lambda_h^{mn}\}_{m=1,\cdots,N;\ (m,n)\in\Xi}$. *(2) Let* $u_h(P) = u_h^m(P)$ *for* $P \in \Omega^m$. *Then* u_h *is the solution of the conventional finite element approximation:*

Find $u_h \in X_h(\Omega)$ *such that*

$$\int_\Omega (\bar{u}_h \bar{v}_h + d\nabla u_h \nabla v_h) = \int_\Omega \bar{f}_h \bar{v}_h \quad \forall v_h \in X_h(\Omega). \tag{3.3}$$

Although solutions of a parallel code of type (2) are not the same as those of (3.3) in general as stated at the beginning of this section, Proposition 3.1 holds for (3.2) since (3.2) is conforming in the sense that the meshes match at subdomains' boundaries. On the other hand, solutions do not always agree between a conventional serial scheme and a non-conforming domain decomposition method, like the mortar element method ([1]) which has the advantage to permit non-matching non-overlapping finite elements at interfaces of subdomains.

4 Matrix representation

Denoting the lumped mass matrix and the stiffness matrix in Ω^m by $\mathbf{M}_m$ and $\mathbf{K}_m$, respectively, we put $\mathbf{A}_m = \mathbf{M}_m + d\mathbf{K}_m$, $m = 1, \cdots, N$. For $(m,n) \in \Xi$ the lumped mass matrix on Γ^{mn} is denoted by $-\mathbf{B}_{mn}$, and $\mathbf{R}_{mn}$ is the matrix representation of the restriction operator $\bar{R}_{mn} : X_h(\Omega^m) \to L^2(\Gamma^{mn})$ defined by $\bar{R}_{mn} v_h^m(P) = \bar{v}_h^m(P)$ for $P \in \Gamma^{mn}$. We thus obtain the following matrix representation of (3.2):

Find vectors U^m and Λ^{mn}, $m = 1, \cdots, N$ and $(m,n) \in \Xi$, such that

$$\begin{aligned}
&\mathbf{A}_m \mathrm{U}^m + \textstyle\sum_{(m,n)\in\Xi} \mathbf{R}_{mn}^T \mathbf{B}_{mn}^T \Lambda^{mn} = \mathbf{M}_m \mathrm{F}^m, \qquad m = 1, \cdots, N, \\
&\mathbf{B}_{mn}(\mathbf{R}_{mn}\mathrm{U}^m - \mathbf{R}_{nm}\mathrm{U}^n) = 0, \qquad (m,n) \in \Xi, \\
&\Lambda^{mn} + \Lambda^{nm} = 0, \qquad (m,n) \in \Xi,
\end{aligned} \tag{4.1}$$

restrictive condition on Λ^{mn}'s corresponding to (*).

Here, U^m, Λ^{mn} and F^m are the matrix representation of u_h^m, λ_h^{mn} and the restriction of f_h on Ω^m, respectively. We note that both $\mathbf{M}_m$ and $\mathbf{B}_{mn}$ are diagonal by virtue of the mass lumping. We fix a subset Ξ_0 of Ξ so that for each $(m,n) \in \Xi$ exactly one of (m,n) and (n,m) belongs to Ξ_0, and denote by Λ the column vector consisting of Λ^{mn}, $(m,n) \in \Xi_0$. We also introduce the diagonal matrix $\mathbf{M} = (\mathbf{M}_1, \cdots, \mathbf{M}_N)^T$ and the following block diagonal matrix $\mathbf{A}$ (one block per subdomain) and column vectors U and F:

$$\mathbf{A} = \begin{pmatrix} \mathbf{A}_1 & & \\ & \ddots & \\ & & \mathbf{A}_N \end{pmatrix}, \quad \mathrm{U} = \begin{pmatrix} \mathrm{U}^1 \\ \vdots \\ \mathrm{U}^N \end{pmatrix} \quad \text{and} \quad \mathrm{F} = \begin{pmatrix} \mathrm{F}^1 \\ \vdots \\ \mathrm{F}^N \end{pmatrix}.$$

Then (4.1) can be rewritten as

Find vectors U and Λ such that

$$\mathbf{A}U + \mathbf{R}^T\mathbf{B}^T\Lambda = \mathrm{MF}, \qquad \mathbf{BR}U = 0 \tag{4.2}$$

restrictive condition on Λ's corresponding to (*).

In the case where the rectangle Ω is decomposed into four rectangles as shown in Figure 4.1 and Λ is fixed as $\Lambda = (\Lambda^{12}\ \Lambda^{23}\ \Lambda^{34}\ \Lambda^{41})^T$, the matrices $\mathbf{R}$ and $\mathbf{B}$ are given respectively by

$$\mathbf{R} = \begin{pmatrix} \mathbf{R}_{12} & & & \\ & \mathbf{R}_{23} & & \\ & & \mathbf{R}_{34} & \\ & & & \mathbf{R}_{41} \\ & \mathbf{R}_{21} & & \\ & & \mathbf{R}_{32} & \\ & & & \mathbf{R}_{43} \\ \mathbf{R}_{14} & & & \end{pmatrix} \quad \text{and}$$

$$\mathbf{B} = \begin{pmatrix} \mathbf{B}_{12} & & & -\mathbf{B}_{12} & & & \\ & \mathbf{B}_{23} & & & -\mathbf{B}_{23} & & \\ & & \mathbf{B}_{34} & & & -\mathbf{B}_{34} & \\ & & & \mathbf{B}_{41} & & & -\mathbf{B}_{41} \end{pmatrix}.$$

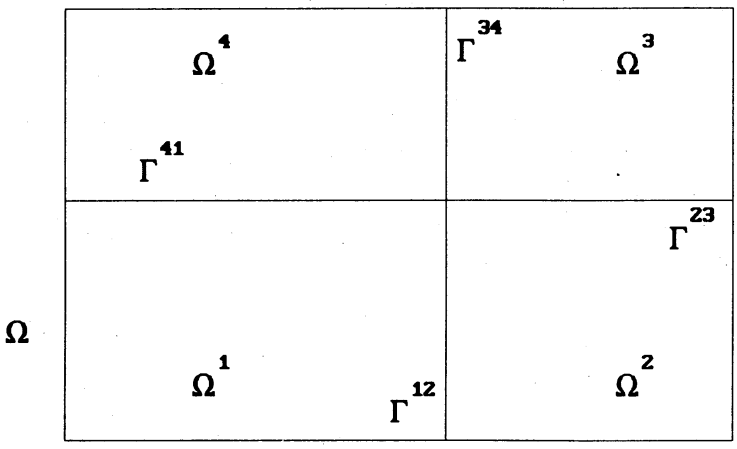

Figure 4.1: Domain decomposition into 4 subdomains

5 Preconditioned conjugate gradient method

Rewriting the first equation in (4.2) as $\mathrm{U} = \mathbf{A}^{-1}(\mathrm{MF} - \mathbf{R}^T\mathbf{B}^T\Lambda)$ and substituting this into the second one, we obtain the following system of linear equations with respect to Λ alone:

$$\mathbf{BRA}^{-1}\mathbf{R}^T\mathbf{B}^T\Lambda = \mathbf{BRA}^{-1}\mathrm{MF}, \qquad \text{restrictive condition on } \Lambda \text{ corresponding to (*).} \tag{5.1}$$

Our scheme solves the linear system (5.1) by a preconditioned conjugate gradient method on a parallel computer. Let us explain our preconditioner here. In the case of the domain decomposition in Figure 4.1 for instance, $\mathbf{RAR}^T$ is expressed as

$$\mathbf{RAR}^T = \begin{pmatrix} \mathbf{A}_{122} & & & & & & & \mathbf{A}_{124} \\ & \mathbf{A}_{223} & & & \mathbf{A}_{231} & & & \\ & & \mathbf{A}_{344} & & & \mathbf{A}_{342} & & \\ & & & \mathbf{A}_{411} & & & \mathbf{A}_{413} & \\ & \mathbf{A}_{213} & & & \mathbf{A}_{211} & & & \\ & & \mathbf{A}_{324} & & & \mathbf{A}_{322} & & \\ & & & \mathbf{A}_{431} & & & \mathbf{A}_{433} & \\ \mathbf{A}_{142} & & & & & & & \mathbf{A}_{144} \end{pmatrix}, \tag{5.2}$$

where $\mathbf{A}_{mnk} = \mathbf{R}_{mn}\mathbf{A}_m\mathbf{R}_{mk}^T$. We construct a preconditioner by noting (5.2). For $m = 1, \cdots, N$ let $\tilde{\mathbf{A}}_m = (\tilde{a}_{ij})$ be a matrix obtained by deforming $\mathbf{A}_m = (a_{ij})$ as follows:

- if a row or column number i corresponds to a cross-point produced by the domain decomposition, then we put $\tilde{a}_{ii} = a_{ii}$ and $\tilde{a}_{ij} = \tilde{a}_{ji} = 0$ $(i \neq j)$,
- if a row number i corresponds to a vertex belonging to $\Gamma^m \backslash \{\text{cross-points}\}$, then we put $\tilde{a}_{ii} = a_{ii} + \sum_j a_{ij}$, where the summation covers column numbers j corresponding to vertices in $\Omega^m \setminus \Gamma^m$.

We thus introduce a block diagonal matrix $\tilde{\mathbf{A}}$ (two blocks per internal interface Γ^{mn}) consisting of $\tilde{\mathbf{A}}_{mnn} = \mathbf{R}_{mn}\tilde{\mathbf{A}}_m\mathbf{R}_{mn}^T$, $(m,n) \in \Xi$, and utilize $\tilde{\mathbf{A}}^{-1}$ as an approximation of $\mathbf{RA}^{-1}\mathbf{R}^T$ in the precondition process in our scheme. For the domain decomposition in Figure 4.1, $\tilde{\mathbf{A}}$ is given by

$$\tilde{\mathbf{A}} = (\tilde{\mathbf{A}}_{122}\ \tilde{\mathbf{A}}_{233}\ \tilde{\mathbf{A}}_{344}\ \tilde{\mathbf{A}}_{411}\ \tilde{\mathbf{A}}_{211}\ \tilde{\mathbf{A}}_{322}\ \tilde{\mathbf{A}}_{433}\ \tilde{\mathbf{A}}_{144}) \qquad \text{(block diagonal)}, \tag{5.3}$$

which approximates (5.2) in some sense.

A usual algorithm of preconditioned conjugate gradient method for (5.1) is stated as below:

(parallel computation) solve the system of linear equations $\mathbf{AG} = \mathrm{MF}$
(parallel computation and data communication) calculate $\mathrm{g} = \mathbf{BRG}$

(parallel computation and data communication)
solve $\mathbf{B}\tilde{\mathbf{A}}^{-1}\mathbf{B}^T\mathrm{y} = \mathrm{g}$ and calculate the inner product $\mathrm{y}^T\mathrm{g}$
(parallel computation) give the initial guess Λ_0
(data communication and parallel computation)
solve the system of linear equations $\mathbf{A}\mathrm{U}_0 = \mathbf{M}\mathrm{F} - \mathbf{R}^T\mathbf{B}^T\Lambda_0$
(parallel computation and data communication) calculate $\mathrm{r}_0 = \mathbf{B}\mathbf{R}\mathrm{U}_0$
(parallel computation and data communication)
solve $\mathbf{B}\tilde{\mathbf{A}}^{-1}\mathbf{B}^T\mathrm{p}_0 = \mathrm{r}_0$ and calculate $\rho = \mathrm{p}_0^T\mathrm{r}_0$
for $k = 0, 1, 2, \cdots$ until $\rho < \epsilon \mathrm{y}^T\mathrm{g}$
(ϵ : a fixed positive constant giving the stopping criterion)
begin
(data communication and parallel computation)
solve the system of linear equations $\mathbf{A}\mathrm{W} = \mathbf{R}^T\mathbf{B}^T\mathrm{p}_k$
(parallel computation and data communication) calculate $\mathrm{q} = \mathbf{B}\mathbf{R}\mathrm{W}$
(data communication) calculate $\theta = \mathrm{p}_k^T\mathrm{q}$ and put $\alpha = \rho/\theta$
(parallel computation)
calculate $\Lambda_{k+1} = \Lambda_k + \alpha\mathrm{p}_k$, $\mathrm{r}_{k+1} = \mathrm{r}_k - \alpha\mathrm{q}$ and $\mathrm{U}_{k+1} = \mathrm{U}_k - \alpha\mathrm{W}$
(parallel computation and data communication)
solve $\mathbf{B}\tilde{\mathbf{A}}^{-1}\mathbf{B}^T\mathrm{s} = \mathrm{r}_{k+1}$, calculate $\mu = \mathrm{s}^T\mathrm{r}_{k+1}$ and $\beta = \mu/\rho$, and put $\rho = \mu$
(parallel computation) calculate $\mathrm{p}_{k+1} = \mathrm{s} + \beta\mathrm{p}_k$
end

In the procedures, one needs to take care of the restrictive condition on Λ corresponding to (*) in Section 3 only when one solves the linear system $\mathbf{B}\tilde{\mathbf{A}}^{-1}\mathbf{B}^T\mathrm{p} = \mathrm{r}$, which has a unique solution due the restrictive condition. The above algorithm, however, requires frequent data communication between subdomains. We note that the inner product $\mathrm{p}^T\mathrm{r}$ equals to $\hat{\mathrm{p}}^T\check{\mathrm{r}}$ if $\hat{\mathrm{p}} = \mathbf{B}^T\mathrm{p}$ and $\mathrm{r} = \mathbf{B}\check{\mathrm{r}}$ ([2]). By using this feature, we can reduce very much the frequency and amount of data communication between subdomains. The final form of our algorithm is now given by

(parallel computation) solve the system of linear equations $\mathbf{A}\mathrm{G} = \mathbf{M}\mathrm{F}$
(parallel computation) calculate $\check{\mathrm{g}} = \mathbf{R}\mathrm{G}$
(parallel computation and data communication)
solve $\mathbf{B}\tilde{\mathbf{A}}^{-1}\hat{\mathrm{g}} = \mathbf{B}\check{\mathrm{g}}$ and calculate the inner product $\hat{\mathrm{g}}^T\check{\mathrm{g}}$
(data communication) give the initial guess Λ_0 and calculate $\hat{\Lambda}_0 = \mathbf{B}^T\Lambda_0$
(parallel computation) solve the system of linear equations $\mathbf{A}\mathrm{U}_0 = \mathbf{M}\mathrm{F} - \mathbf{R}^T\hat{\Lambda}_0$
(parallel computation) calculate $\check{\mathrm{r}}_0 = \mathbf{R}\mathrm{U}_0$
(parallel computation and data communication)
solve $\mathbf{B}\tilde{\mathbf{A}}^{-1}\hat{\mathrm{p}}_0 = \mathbf{B}\check{\mathrm{r}}_0$ and calculate $\rho = \hat{\mathrm{p}}_0^T\check{\mathrm{r}}_0$
for $k = 0, 1, 2, \cdots$ until $\rho < \epsilon\hat{\mathrm{g}}^T\check{\mathrm{g}}$
(ϵ : a fixed positive constant giving the stopping criterion)
begin
(parallel computation) solve the system of linear equations $\mathbf{A}\mathrm{W} = \mathbf{R}^T\hat{\mathrm{p}}_k$
(parallel computation) calculate $\check{\mathrm{q}} = \mathbf{R}\mathrm{W}$

(data communication) calculate $\theta = \hat{\mathrm{p}}_k^T \breve{\mathrm{q}}$ and put $\alpha = \rho/\theta$
(parallel computation)
calculate $\hat{\Lambda}_{k+1} = \hat{\Lambda}_k + \alpha \hat{\mathrm{p}}_k$, $\breve{\mathrm{r}}_{k+1} = \breve{\mathrm{r}}_k - \alpha \breve{\mathrm{q}}$ and $\mathrm{U}_{k+1} = \mathrm{U}_k - \alpha \mathrm{W}$
(parallel computation and data communication)
solve $\mathbf{B}\tilde{\mathbf{A}}^{-1}\hat{\mathrm{s}} = \mathbf{B}\breve{\mathrm{r}}_{k+1}$, calculate $\mu = \hat{\mathrm{s}}^T \breve{\mathrm{r}}_{k+1}$ and $\beta = \mu/\rho$, and put $\rho = \mu$
(parallel computation) calculate $\hat{\mathrm{p}}_{k+1} = \hat{\mathrm{s}} + \beta \hat{\mathrm{p}}_k$
end

References

[1] Y. Achdou and Yu. A. Kuznetsov, Algorithm for a non conforming domain decomposition method, ftp://barbes.polytechnique.fr/pub/RI/1994/, 1994.
[2] S. Fujima, On parallel computation by mortar element method, Abstract of the 45th Nat. Cong. of Theoretical and Applied Mechanics, pp. 55-56, 1996 (in Japanese).
[3] T. Ikeda, Numerical simulation for nonlinear oscillation of internal layers, Lecture Notes in Numerical and Applied Analysis Vol. 14 (Advances in NUmerical Mathematics: Proceedings of the 2nd Japan-China Seminar on Numerical Mathematics, eds. T. Ushijima et al.), pp. 69-77, Kinokuniya, 1995.
[4] A. Suzuki, Fast Stokes Solver based on domain decomposition methods, Abstract of the 46th Nat. Cong. of Theoretical and Applied Mechanics, pp. 235-236, 1997 (in Japanese).